AFRL-VA-WP-TP-2007-315

DECENTRALIZED CONTROL USING GLOBAL OPTIMIZATION (DCGO) (POSTPRINT)

Matthew Flint, Tanya Khovanova, and Michael L. Curry

BAE Systems

MARCH 2007

**AIR FORCE RESEARCH LABORATORY
AIR VEHICLES DIRECTORATE
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542
AIR FORCE MATERIEL COMMAND
UNITED STATES AIR FORCE**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Wright Site (AFRL/WS) Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-VA-WP-TP-2007-315 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

*//Signature//
DANIEL J. SCHREITER
Project Engineer

//Signature//
MICHAEL P. CAMDEN
Chief, Control Systems Development and
Application Branch
Control Sciences Division

//Signature//
JEFFREY C. TROMP
Senior Technical Advisor
Control Sciences Division
Air Vehicles Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

*Disseminated copies will show "//Signature//" stamped or typed above the signature blocks.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YY)*	**2. REPORT TYPE**	**3. DATES COVERED *(From - To)***
March 2007	Conference Paper Postprint	01 October 2002 – 01 April 2005

4. TITLE AND SUBTITLE
DECENTRALIZED CONTROL USING GLOBAL OPTIMIZATION (DCGO) (POSTPRINT)

5a. CONTRACT NUMBER
F33615-02-C-3262

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0602201

6. AUTHOR(S)
Matthew Flint and Tanya Khovanova (BAE Systems)
Michael L. Curry (Raytheon)

5d. PROJECT NUMBER
A013

5e. TASK NUMBER

5f. WORK UNIT NUMBER
0A

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

BAE Systems
6 New England Executive Park
Burlington, MA 01803

Raytheon
50 Apple Hill Drive
Tewksbury, MA 01876

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory
Air Vehicles Directorate
Wright-Patterson Air Force Base, OH 45433-7542
Air Force Materiel Command
United States Air Force

10. SPONSORING/MONITORING AGENCY ACRONYM(S)
AFRL/VACC

11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)
AFRL-VA-WP-TP-2007-315

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES
Conference paper published in the Proceedings of the 2007 Conference and Exhibit, published by AIAA.

This work was funded in whole or in part by Department of the Air Force contract F33615-02-C-3262. The U.S. Government has for itself and others acting on its behalf an unlimited, paid-up, nonexclusive, irrevocable worldwide license to use, modify, reproduce, release, perform, display, or disclose the work by or on behalf of the U.S. Government. PAO Case Number: AFRL/WS 07-0589 (cleared March 16, 2007). Paper contains color.

14. ABSTRACT

The coordination of a team of distributed air vehicles requires a complex optimization, balancing limited communication bandwidths, non-instantaneous planning times and network delays, while at the same time trying to allocate limited resources to spatially diverse locations in a near-optimal fashion in a dynamic and uncertain environment. Given that, in this environment, the optimality of a given plan will not last very long when the information state is constantly changing and being updated, a new approach is proposed in this paper. Global-scope plans for the team are generated and distributed using the principle of emergent leadership to provide efficient plan generation and execution with minimal performance degradation compared to a centralized controller under delayed communications. This type of protocol is labeled the Decentralized Control Global Optimization (DCGO) protocol, and is discussed in this paper, along with some simulation results showing that this premise can produce good results in a realistic environment.

15. SUBJECT TERMS
Decentralized control, global optimization, emergent leadership, team formation, plan execution

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Daniel J. Schreiter
Unclassified	Unclassified	Unclassified	SAR	18	**19b. TELEPHONE NUMBER** *(Include Area Code)* N/A

AIAA <i>Infotech@Aerospace</i> 2007 Conference and Exhibit
7 - 10 May 2007, Rohnert Park, California

AIAA 2007-2906

Decentralized Control Using Global Optimization (DCGO)

Matthew Flint* and Tanya Khovanova†
BAE Systems, Burlington, MA, 01803, USA

Michael L. Curry‡
Raytheon Company, Andover, MA, 01810, USA

The coordination of a team of distributed air vehicles requires a complex optimization, balancing limited communication bandwidths, non-instantaneous planning times and network delays, while at the same time trying to allocate limited resources to spatially diverse locations in a near-optimal fashion in a dynamic and uncertain environment. Given that, in this environment, the optimality of a given plan will not last very long when the information state is constantly changing and being updated, a new approach is proposed in this paper. Global-scope plans for the team are generated and distributed using the principle of *emergent leadership* to provide efficient plan generation and execution with minimal performance degradation compared to a centralized controller under delayed communications. This type of protocol is labeled the *Decentralized Control Global Optimization* (DCGO) protocol, and is discussed in this paper, along with some simulation results showing that this premise can produce good results in a realistic environment.

I. Introduction

Unmanned aerial vehicles have been receiving a large amount of attention in recent research literature. Their relatively low-cost and high endurance compared to manned aircraft makes them useful in many situations. However, current unmanned vehicles are expensive to operate in environments where they must interact with manned or other unmanned systems, because the typical operation of these aircraft require the oversight of one or more humans per vehicle, who are often piloting, handling communications, targeting, etc. Coordinating between these disparate groups of individuals can be quite time consuming. This type of environment has a very fast time scale, so tight coordination is necessary for successfully accomplishing tasks within it. In such situations, it is important to ensure that every vehicle performs the actions that are most appropriate (in terms of assets, position, fuel, and so on) to that vehicle within coordinated bands of time so that other team members can also accomplish their actions within the integrated threat environment. This coordination can be accomplished using autonomous planning, where the vehicles themselves are able to create plans that they or other vehicles can follow, based on the best information available to them. These plans should be global-scope plans, in which assignments for every agent (e.g. a UAV) are explicitly included. This is as opposed to a local-scope plan, which contains assignments for a single vehicle, or some set of vehicles smaller than the entire team. This is not to say that an operator has been taken out of the control loop, either, as once the plans have been created, they can be approved by an operator. However, this adds certain delays in the system, and are the subject of ongoing research, and will be considered in future work.

Traditionally, global-scope planning has been done using a centralized planner, where the centralization is in terms of location (planning for all agents is done in one single location) and information state (information from each vehicle is immediately included in the plan). (See e.g.[1] [2] [3] [4] [5] [6] [7] [8]) It is possible to use a centralized controller with decentralized execution, where this means that the planner controls the actions of decentralized (again, in terms of both location and information state) vehicles across a distributed

[*] Lead Research Engineer, Advanced Information Technologies, 6 New England Executive Park, Burlington, MA, 01803, AIAA member. Email: matt.flint@baesystems.com

[†] Lead Research Engineer, Advanced Information Technologies, 6 New England Executive Park, Burlington, MA, 01803.

[‡] Senior System Engineer, System Architecture, Design, & Integration Directorate, 50 Apple Hill Drive Tewksbury, MA 01876

communication and computational environment. However, modern unmanned vehicles are capable of not only the execution of a plan, but of also creating them. Thus, the problem can be decentralized, often removing certain restrictive assumptions on communication and planning that are often difficult to meet in a realistic environment.

In most realistic scenarios, information appears asynchronously based upon local sensor collections and the response time to this information is of critical importance. Thus, the information state available to any planner has to be necessarily decentralized because communications may not be fast enough to send information to a central location, wait for the plan to be generated, and then get the plan back.

One solution to this problem is to decentralize the planning, such that each vehicle is able to create a plan, based on local information. Ideas such as this exist in recent literature (e.g.[9] [10] [11] [12]), with certain assumptions about information availability and communication constraints. The challenge then becomes retaining the tight coordination, and therefore the performance, of a global-scope (centralized) controller in this decentralized environment. Some work has been started in this area, as in references ([13] [14] [15] [16]). This can be similar in nature to a database synchronization problem (see e.g.[17] [18]), although this work generally does not fit the particular assumption set that UAV problems typically have.

After examining the previous work, this paper proposes a novel new approach to this problem. It introduces a protocol called the Decentralized Control Global Optimization (DCGO) protocol that addresses this problem. The "Global" refers to the fact that global level planning is being done even in a decentralized (in terms of location and information state) environment.

The DCGO protocol makes use of a principle called *Emergent Leadership.* Under this principle, the decision of which vehicle will create the global scope plan (i.e. leadership) that all team members will follow is not decided *a priori*; instead leadership of the team is decided dynamically and potentially even after plans have been generated and transmitted over the communications network. Emergent Leadership is implemented following these steps:

Generate Plan: a vehicle creates a plan based on available information.

Select Plan: Since leadership is not determined *a priori*, there may be multiple plans in the system.

Execute Plan: Once the plan to execute has been selected, that plan is executed by all vehicles in the team.

The algorithms used to generate the plan are not discussed in this paper, as the protocol has been formulated to be useful for any global-scope plan generation algorithm, where the plan consists of vehicle resource-task pairings along with routes to follow to execute the mission. It is assumed that plan-generation algorithms can be devised that are appropriate to the computation and time requirements available to the vehicles for any particular scenario.

This paper is organized as follows: The problem formulation is given in Section II. The protocol is given in detail in Section III. Some theoretical results proving that this approach is superior to a fixed (or default) leader case is given in Section IV. Some brief simulation results demonstrating that this is a feasible approach are given in Section V. Lastly, conclusions and a bibliography are given.

II. Problem Formulation

The problem details are specified in this section. First, it is necessary to give some definitions. Next, the assumptions used in developing the protocol are given, including the assumptions about what information is available at the beginning of a mission and during the mission.

II.A. Definitions

Plan: A set of vehicles' schedules, where a schedule consists of a sequence of activity points, each of which consists of a description of an activity and a time of execution. Executing the plan should produce a successful mission.

Winning Plan: A plan that will be executed by all members of a team.

Leader: The vehicle that generates a winning plan. A vehicle can potentially transition between a leader, a potential leader and a follower during a mission.

Potential Leader: A vehicle that generates a plan, when it is not known if this plan will win. A vehicle can potentially transition between a leader, a potential leader and a follower during a mission.

Follower: A vehicle that knows that it is not currently the leader. A vehicle can potentially transition between a leader, a potential leader and a follower during a mission.

Triggering Event: An event (or a piece of information) that must be incorporated into a new plan.

Execution time of the plan: The time at which a plan is scheduled to begin execution.

Planning time: The time required to generate a new plan.

Communications delay: The amount of time it takes for a message to be received by one vehicle after it is sent by another, or the time it takes an exogenous broadcast to reach all the vehicles.

Response time: Total amount of time between the detection of a triggering event and the execution time of a plan that incorporates that event.

II.B. Vehicle Assumptions

- Clocks on all vehicles are synchronized.
- There is no attrition among the air vehicles. For the purpose of analyzing and demonstrating DCGO capabilities, the possibility of loss of a vehicle was assumed to be negligible.
- Every vehicle has the capability to generate a plan.
- Exact vehicles' locations can be calculated given the plan and current vehicle velocities (by projecting the locations forward in time).
- There are no communications failures. It is assumed that all messages will eventually be received (i.e. with a finite delay).
- Communication delays are not longer than some fixed number - T_{MCD} (MCD: Maximum Communication Delay).
- The planning time is not longer than some fixed number - T_{MPT} (MPT: Maximum Planning Time).

II.C. New information appearing during the mission

Since the mission is taking place in a dynamic environment, new information can appear in the course of the mission. For the purposes of this protocol, the information is assumed to be one of two categories:

Triggering Event – A piece of information that is important enough or causes enough of a disruption to the current winning plan that it must be included in a newly generated plan.

Non-triggering event – Any information receiving during the mission that does not fall into the above category.

III. The protocol

Every vehicle is a planner; since there are multiple planners in the system, the cost of any one of them starting a plan is relatively inexpensive. A guiding principle for this approach is that it is better to generate more plans than are needed and then to discard some of them, rather than wait for synchronization before generating a plan that will be executed. This allows the response time to detected events to be minimized, compared with other, less advanced approaches.

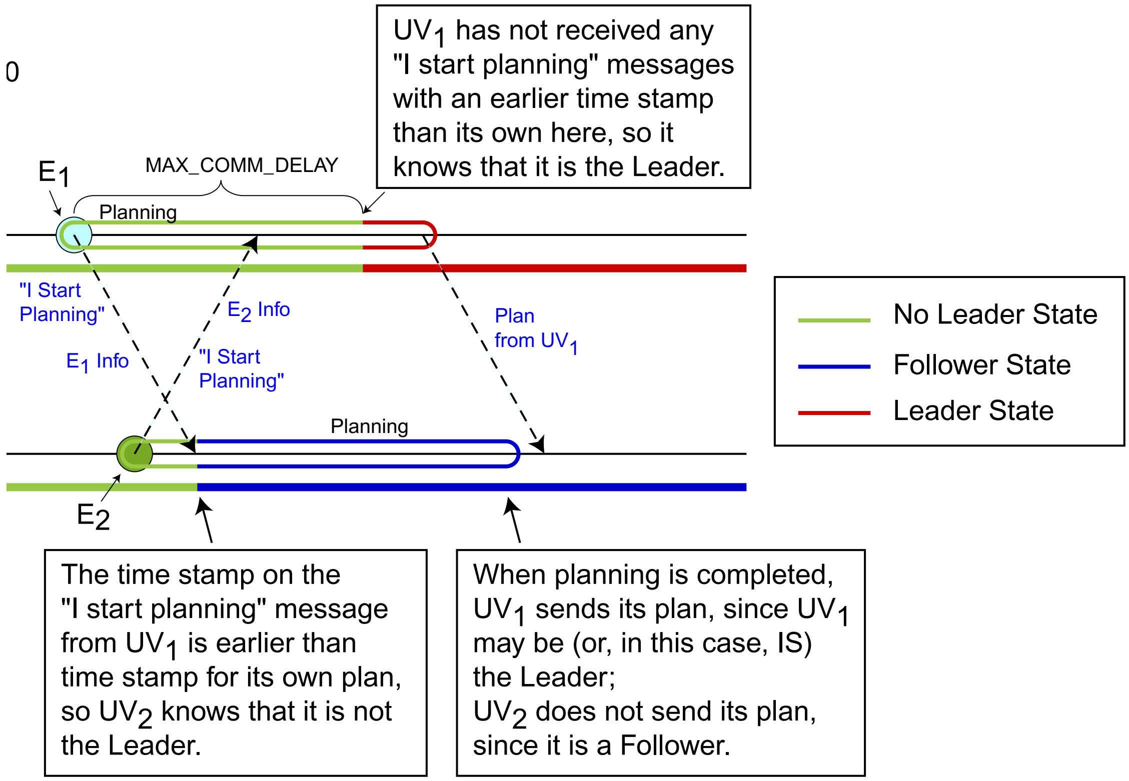
0
UV$_1$ has not received any "I start planning" messages with an earlier time stamp than its own here, so it knows that it is the Leader.
E$_1$
MAX_COMM_DELAY
Planning
"I Start Planning"
E$_2$ Info
E$_1$ Info
"I Start Planning"
Plan from UV$_1$
Planning
E$_2$
No Leader State
Follower State
Leader State
The time stamp on the "I start planning" message from UV$_1$ is earlier than time stamp for its own plan, so UV$_2$ knows that it is not the Leader.
When planning is completed, UV$_1$ sends its plan, since UV$_1$ may be (or, in this case, IS) the Leader;
UV$_2$ does not send its plan, since it is a Follower.

Figure 1. This figure shows the basic structure of the message transfer and leader state transitions of the DCGO protocol. Both vehicles start out as potential leaders, but after coordination, UV1 becomes the leader because it can react to information the fastest.

III.A. Overview of the protocol

In developing the protocol the following goals were considered primary:

1. Events are not dropped. Any triggering event is incorporated into all subsequently generated plans.
2. Fast response, i.e. minimize the maximum response time.
3. All vehicles are executing the same plan at all times.

When a triggering event is received, the information about the event is circulated, the plans (once generated) are broadcast to the rest of the team, a leader is determined, and then the winning plan is executed at the plan execution time. Due to variances in the delays involved, these things may happen in different orders in different cases. The leader remains “in office” until the last plan the leader created is executed. A consequence of the DCGO protocol is that events that arrive very close to each other are all planned by the same leader. If triggering events are sparse, the protocol creates a distributed controller situation, where the most suitable vehicle becomes the replanning leader.

In order to take advantage of inter-vehicle cooperation, team vehicles execute a common plan. However, in a dynamic environment, it is assumed that at some point a triggering event occurs. Thus, some vehicle must generate a plan that incorporates this triggering event, and then share this updated plan. What follows is an explanation of how the decision is made regarding which vehicle will create this plan. Note that the decision of which plan will win is not always made before the planning process is begun: in fact, many vehicles may be planning; and many plans may be shared. Because, in this case, minimizing the response time is one of the primary goals, timing rules are used to determine which plan is the one that will be executed.

Intra-team Messages: The decision regarding which plan is the winning plan is based on time stamp values related to communications shared among team members. In the DCGO protocol, it is assumed that several kinds of messages are sent to all team members by each vehicle as required. Every message has a time stamp.

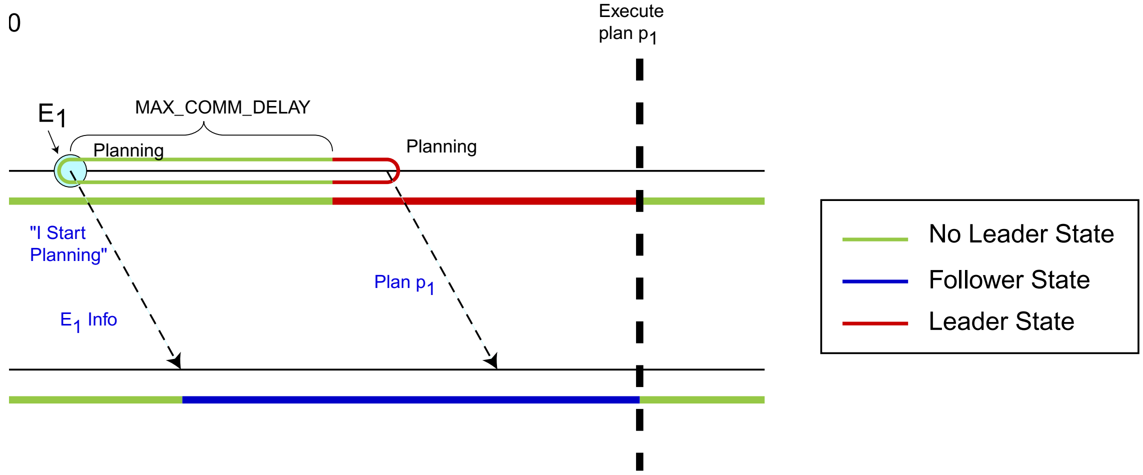

Figure 2. Once a vehicle becomes the leader, it stays the leader until the minimum leader switching time has elapsed. This occurs at the execution time of the latest plan, after which all other vehicles are guaranteed to know the predicted future state.

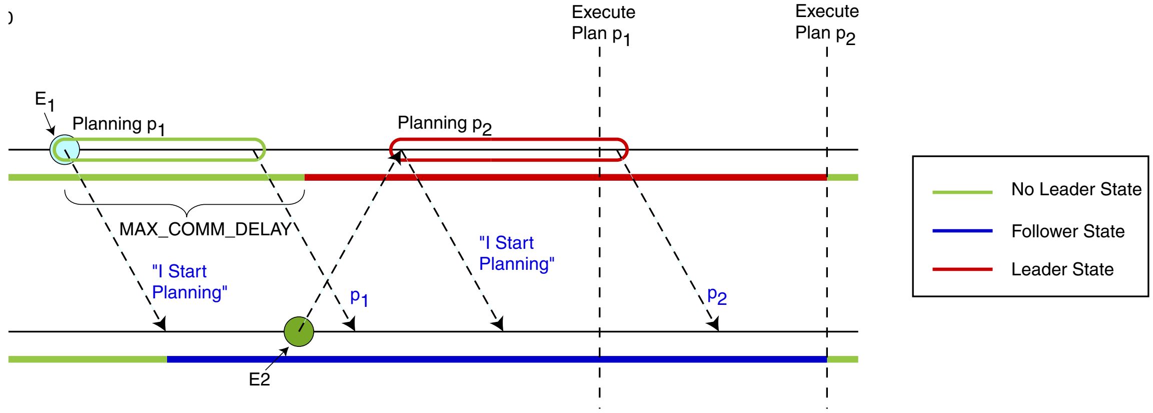

Figure 3. Once UV1 becomes the leader, if more information is received, it stays the leader. This is because the leader has access to the predicted future state before any other vehicle until the minimum leader switch time has elapsed.

- Triggering Event Information: The time of an event, other event-specific information, and the vehicle which detected the event are shared by all team members immediately after the event is detected.
- ”I Start Planning”: All vehicles need to know that the Leader (or a potential Leader, in the case where the Leader has not yet been determined) is planning, so the Leader (or each potential leader) sends an ”I Start Planning” message at the moment it starts planning.
- New Plan: At the end of replanning the plan is sent out immediately unless it is already known that this plan is not the winner.

III.B. Design decisions and considerations

The protocol design decisions can be categorized into several section:

III.B.1. Information dissemination

As soon as a vehicle receives any new triggering events, it sends this information to all other vehicles. At time T, each vehicle is guaranteed to have all the information on the events which happened before time T - T_{MCD}. Similarly, when events are generated exogenously, it is assumed that all vehicles will receive the event within T_{MCD} of the exogenous broadcast.

III.B.2. Interruption

If planners were allowed to interrupt planning because of new triggering events, it may be possible that no plan will ever be finished (or executed). This can occur if triggering events are coming in fast enough that a new plan would continuously be started without finishing the previous one. Hence, in this design if a vehicle starts planning it always finishes the plan. However, in some cases, the finished plan may be discarded.

III.B.3. Execution time delay

The new plan must arrive at each vehicle no later than the plan execution time. This is guaranteed by setting "execution time delay", the difference between the execution time of a plan and the time new plan generation had been triggered, to be not less than T_{MPT} plus T_{MCD}, in order to ensure enough time to create this plan and make sure that all vehicles have received it. To minimize the response time the execution time delay is set equal to $T_{MPT} + T_{MCD}$. Note that the execution time of the plans is determined when the generation of the plan begins.

III.B.4. Potential future plans list

Vehicles could receive several plans before they determine the leader. Each vehicle maintains all plans until it is determined which plan is the winning plan. Non-winning plans are discarded. At a given time, it is possible that there is more than one winning plan, each with a different future execution time. These will be executed sequentially, beginning at the respective execution times.

III.B.5. Predicted future state

When a vehicle plans, it needs to take into account that some activities will be performed while it is planning and while the plan is being transferred. As the vehicles fly with known speeds and trajectories, given the initial location of vehicles and their schedules, it is easy to calculate where all the vehicles will be at any point in time. Also, it is possible to calculate what activities will be performed and what munitions will be left at some future time. Thus, when a vehicle starts planning, it calculates a snapshot of the future world state at the time where execution of the plan being generated would begin. This snapshot is called the "predicted future state". The winning plan must have been generated using the correct future state, and can be used to predict the next future state. This allows the team of vehicles to execute plans on time.

III.B.6. Future executable plans list

The leader does not need to wait until the execution time of the most recent plan in order to correctly predict the future state. The moment the last winning plan is generated the leader can start to generate the next plan. As a result, it is possible that the leader can generate more than one plan before the execution time of the first plan. To correctly predict the future state, the leader should maintain all the plans that will be executed and generate the predicted future state by taking into account these winning plans.

III.B.7. Minimum Leader Switch Time

As a result of the protocol, there is a minimum amount of time which needs to pass from the moment one vehicle starts generating a winning plan until another vehicle can generate a winning plan. This time is called the "minimum leader switch time". To plan effectively the new leader needs to predict the future state; for that, the latest winning plan should be known. From the moment the Leader begins generating a winning plan, the Leader is the only vehicle guaranteed to know that winning plan until the execution time delay (see above) has passed. Thus, the minimum leader switch time is $T_{MPT} + T_{MCD}$.

III.C. Protocol specifics

1. All the decisions on the winning plan, discarding plans, the vehicle state, etc. are made based on "I Start Planning" messages and their time stamps.

2. When a Leader is not determined, every vehicle starts planning upon receiving a triggering event.

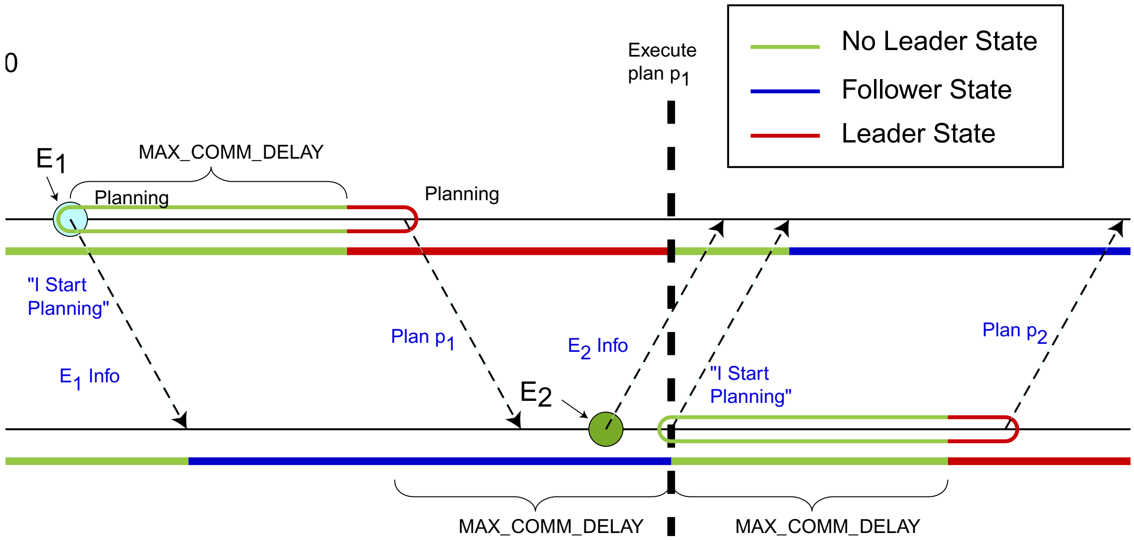

Figure 4. This figure shows the transition of leadership from one vehicle to another. Because of the change in leadership, the team will be able to respond to the event more quickly than if they were using a fixed-leadership protocol.

3. When the Leader is determined, another vehicle can start a winning plan only after the Minimum Leader Switch Time.
4. The Leader is responsible for creating a plan that includes all (not only its) events received before the execution time of its last plan.
5. Before the Leader is determined every vehicle behaves as if it could become the Leader.
6. The winning plan rule: when there is no Leader, the plan with the earliest ”I Start Planning” message wins. In the case where multiple vehicles start planning at the same time, the times of the events that triggered the replan are compared and the earliest one wins. If those times are also equal the unique vehicle IDs are used to break the tie. If there is a plan generated by the Leader after the start of the generation of the last winning plan and before the execution of that plan, this plan wins and the Leader stays the same. If the Leader does not generate a plan before the execution time of its last plan, the vehicles return to a no Leader state.
7. A vehicle does not start planning if it knows that this plan cannot win.
8. At the end of replanning, the plan is sent out immediately unless it can be determined that this plan is not the winner.
9. The vehicles remember the time of their last triggering events. At the moment of executing a new plan, each vehicle that is not a Leader checks if there are triggering events that it owns which might not be incorporated in the plan. These are the events that occurred less than T_{MCD} before the current time. If there are such events and the Leader didn't send any new ”I Start Planning” messages, then the vehicle starts planning and behaves as a potential Leader.

III.D. Some Examples

The figures in this paper demonstrate illuminating cases of the protocol in use. Figure 1 shows the basic leader selection process in the presence of differing local state information on two vehicles. Figure 2 shows the lifecycle of a leader in the case where only one event is detected for a relatively long period. Figure 3 shows a case where a leader is kept in office so that an additional event can be included in a new plan. Lastly, Figure 4 shows a case where leadership is handed off from one vehicle to another, which results in a minimization in the response time to both events.

IV. Results

This section presents results which prove that the goals outlined above are satisfied by the DCGO plan dissemination protocol.

Lemma 1 *All vehicles accept the same plan.*

Proof: The plan that wins is uniquely determined by its time stamp (by the rule described above). The winning plan is always received before the time it should be executed and by the time the decision of winning plan is made. Hence, all vehicles accept the same plan.

Lemma 2 *Each event gets into a plan.*

Proof: The first event gets into a leader's first plan. The response time for this event is T_{MPT} + T_{MCD}. All the events that the leader owns and which happen between the first triggering event and the execution time of the latest current plan are incorporated into a new plan by the leader. If this event occurs while the leader is not planning, the replan starts immediately and the response time is T_{MPT} + T_{MCD}. If the leader is planning during the event time, the response time is not more than 2*T_{MPT} + T_{MCD}. All events that are not owned by the leader and happen between the first triggering event and the execution time of the last leader's plan minus T_{MCD} are incorporated by the leader. In the worst case the event is received during planning and the response time is not more than 2*T_{MPT} + 2*T_{MCD}. All the events that are not owned by the leader and happen after the execution time of the last leader's plan minus T_{MCD} will either be owned by a new leader, in which case the response time is no more than T_{MPT} + 2*T_{MCD}, or will be received by some other new leader in which case the response time is no more than 2*T_{MPT} + 2*T_{MCD}. Since these cases are exhaustive, each event gets into a plan.

Corollary 1 *The maximum response time is 2*T_{MPT} + 2*T_{MCD}.*

Proof: Follows from Lemma 2.

Theorem 1 *The DCGO Plan Dissemination Protocol performs better (by the defined metrics) than the default Leader strategy*

Proof: Like the default leader scheme, Lemmas 1 and 2 show that the DCGO protocol is valid. To show that the DCGO protocol can achieve a faster response time, suppose one of the vehicles is a default Leader for the whole mission, i.e. the Leader does not change. That is, the default leader vehicle starts planning immediately upon receiving any triggering event (the default Leader owns all the events) if it is not currently planning. Then it sends the plan to all the team members. If some new triggering information is received during planning, the default leader starts planning a new plan upon the completion of the previous plan. In this case the response time would be guaranteed to be not more than 2*(T_{MPT} + T_{MCD}). Namely, T_{MCD} is needed to receive information from other vehicles, T_{MPT} is needed to plan, T_{MPT} is needed to wait in case the information is received during planning, and T_{MCD} is needed to send the new plan to other team members. Note that the response time is bounded by the same number for both the default Leader case and the DCGO Plan Dissemination Protocol. This happens in the case where events are happening fast enough (i.e. at a rate faster than the minimum leader switching time) that the DCGO Plan Dissemination Protocol does not switch leaders; hence the response time is the same.

However, in other cases, the DCGO Plan Dissemination Protocol can achieve better results. In the case where triggering events are sparse, the default Leader's response time is bounded by T_{MPT} + 2*T_{MCD} (receive information, plan and send information). DCGO plan dissemination protocol's response time in this case is bounded by T_{MPT} + T_{MCD}, which is always less than the default Leader case. Thus, we conclude that the proposed DCGO protocol has a better overall response time than the default leader protocol.

V. Simulation Results

This coordination protocol was used in a simulation environment using BAE System's proprietary M2CS (multi-vehicle mission control system) planner running in version 1.3 of the Boeing OEP (Open Experimental Platform), an ITAR-(export control-) compliant simulation environment for UAVs. In these tests, 4 aerial

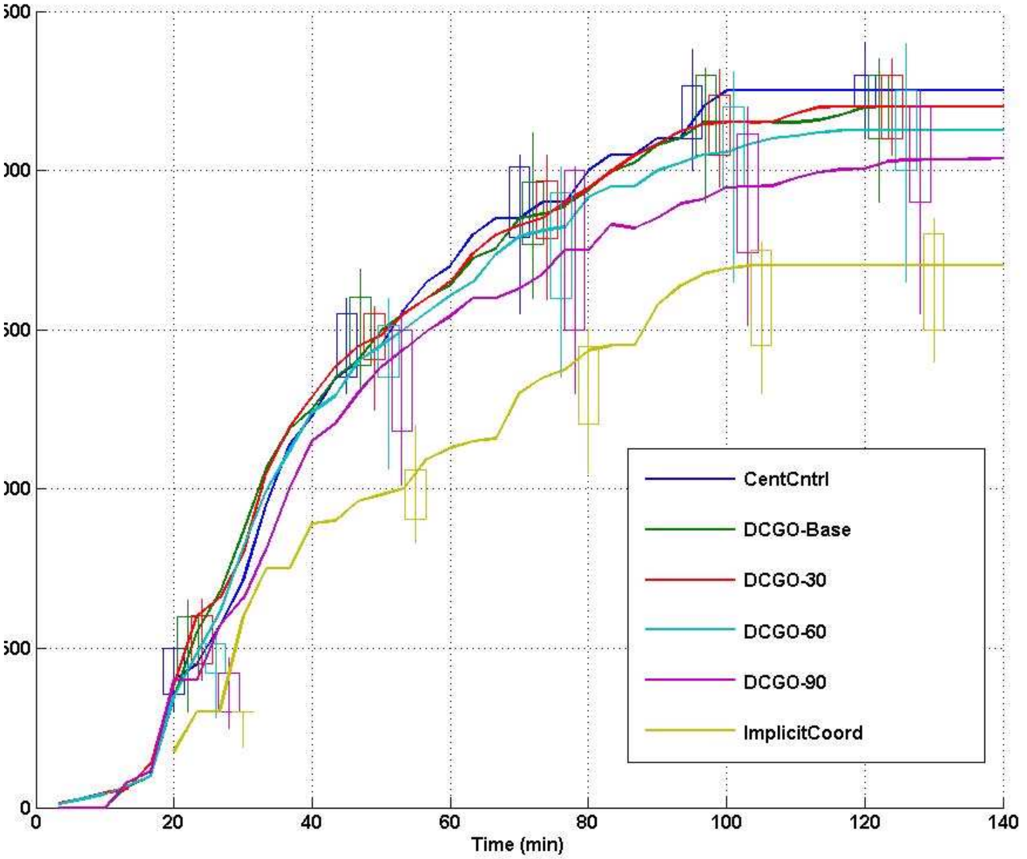

Figure 5. The score achieved by the mission controller utilizing the DCGO protocol, as a function of the time elapsed since the start of the mission. These curves represent the average of several replications of the same scenario with various system delays.

vehicles utilizing identical copies of M2CS are allowed to create plans for a SEAD mission. The coordination between the planners is handled using either an ideal communication channel, the DCGO protocol under varying communication delays, or with "Implicit Coordination" (see below). The delayed information consists of "off board" world state information, such as other vehicle status and target data. The boxes displayed on the curves demonstrate the range of the data collected during the experiment. The results of this simulation are given in Figure 5. These results originally appeared in the technical report for the DCAT program,[19] and have been reproduced here with the permission of the authors.

The curve marked "CentCntrl" represents the results of what occurs when communication is perfect (i.e. no delay). This represents an experimental upper bound, since adding delays to the system can only make it worse. The DCGO-Base and DCGO-30 are the same, but with different replications run. They both utilize the DCGO protocol in the presence of a 30 second delay in information transmitted between vehicles. DCGO-60 and DCGO-90 also utilize the DCGO protocol, at a 60 second and 90 second delay in information transmitted between vehicles, respectively. Lastly, "Implicit coordination" means that the four aerial vehicles all create plans for themselves, with the assumption that all other vehicles are going to generate the same plan, and so each plan is not need to be communicated. (All vehicles are implicitly running the same plan.) In the presence of delays this causes a breakdown in cooperation because what one vehicle assumes about another may not be correct when they each can have different information and the inherent differences in state that result. This provides an experimental example to compare against.

The data provided in Figure 5 shows that the distributed DCGO controllers can achieve near identical results of a centralized controller when world state information is delayed up to 60 seconds, and has reasonable results with information delayed for 90 seconds.

VI. Conclusions and Future Work

What has been shown in this work is a protocol that allows a centralized, global-scope planner, to be used in a decentralized setting. This has many advantages, not least of which is that there exist many very good global-scope planners that can now be used decentrally. This protocol has been shown to be better than a fixed-leader case in terms of response time, and is also extensible in that it can be the foundation for future work.

There are many obvious extension to this work. First, the protocol assumes that vehicle will not be lost during the course of the mission. This is not a realistic assumption in many environments in which UAVs may be expected to operate, however. Introducing a chain-of-command, or back-up, leadership into the protocol can make it robust to losing vehicles during the course of a mission. A further assumption is that message delays are bounded, which precludes the fact that sometimes messages may simply be dropped. This is a challenging problem because it is important to distinguish between a vehicle that is temporarily out of communications and one that has been destroyed. This would probably require each vehicle to have a probabilistic model of the states of team members and react based on this model.

VII. Acknowledgement

This material is based upon work supported by the United States Air Force under Contract No. F33615-02-C-3262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force. This document, AFRL-WS 07-0589, was cleared for public release by AFRL Public Affairs, disposition date 3/16/2007.

The authors would like to acknowledge the contribution provided by John Devereaux, Rob Vrablik, and Jeremy Clarke for their efforts to produce high quality simulation results. The authors would also like to acknowledge the assistance of Cynara Wu, Fred Zeitz, Jerry Wohletz, and David Castañon for their advice on the construction of the paper.

References

[1] Cassandras, C. G. and Li, W., "A Receding Horizon Approach for Solving Some Cooperative Control Problems," *Proc. of the 41st IEEE Conference on Decision and Control*, 2002, pp. 3760–65.

[2] Li, W. and Cassandras, C. G., "Stability Properties of a Cooperative Receding Horizon Controller," *Proc. of the 42st IEEE Conference on Decision and Control*, 2003, pp. 492–97.

[3] Ghose, D., Krichman, M., Speyer, J., and Shamma, J., "Modeling and analysis of air campaign resource allocation: A spatio-temporal decomposition approach," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 32, No. 3, May 2002, pp. 403–418.

[4] Xu, L. and Ü. Özgüner, "Battle Management for Unmanned Aerial Vehicles," *Proc. of the 42nd IEEE Conference on Decision and Control*, 2003, pp. 3585–3590.

[5] Arslan, G., Wolfe, J., Shamma, J., and Speyer, J., "Optimal Planning for Autonomous Air Vehicle Battle Management," *Proc. of the 41st IEEE Conference on Decision and Control*, 2002, pp. 3782–3787.

[6] Kang, W. and Sparks, A., "Modelling and Computation of Optimal Task Assignment for Cooperative Control," *Proc. of the 42st IEEE Conference on Decision and Control*, 2003, pp. 1017–22.

[7] Singh, L. and Fuller, J., "Trajectory Generation for a UAV in Urban Terrain, using Nonlinear MPC," *Proceedings of the 2001 American Control Conference*, 2001, pp. 2301–2308.

[8] Schumacher, C. and Chandler, P. R., "Task Allocation for Wide Area Search Munitions," *Proc. of the American Control Conference*, 2002, pp. 1917–22.

[9] Flint, M., Fernández-Gaucherand, E., and Polycarpou, M., "Stochastic Modeling of a Cooperative Autonomous UAV Search Problem," *Military Operations Research*, Vol. 8, No. 4, 2003, pp. 13–32.

[10] Flint, M., Fernández-Gaucherand, E., and Polycarpou, M., "Cooperative Control for UAV's Searching Risky Environments for Targets," *Proc. of the 42nd IEEE Conference on Decision and Control*, Hyatt Regency Maui, Hawaii, USA, 2003, pp. 3567–3572.

[11] Yang, Y., Minai, A. A., and Polycarpou, M. M., "Decentralized Cooperative Search in UAV's Using Opportunistic Learning," *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, 2002.

[12] Hespanha, J. P., Kim, H. J., and Sastry, S., "Multiple-Agent Probabilistic Pursuit-Evasion Games," *Proc. of the 38th Conf. on Decision and Control*, 1999.

[13] Rudie, K. and Wonham, W., "Think Globally, Act Locally: Decentralized Supervisory Control," *IEEE Transactions on Robotics and Automation*, Vol. 37, No. 11, November 1992, pp. 1692–1708.

[14] Wong, K., Thistle, J., Malhamé, R., and Hoang, H., "Supervisory Control of Distributed Systems: Conflic Resolution," *Proceedings of the 1998 IEEE Conference on Decision and Control*, 1998, pp. 3275–3280.

[15] Nair, R., Tambe, M., and Marsella, S., "Role allocation and reallocation in multiagent teams: Towards a practical analysis," *Second International Joint Conference on Agents and Multiagent Systems (AAMAS)*, 2003, pp. 552–559.

[16] Tabuada, P. and Pappas, G., "Discrete synchronization of hybrid systems," *Proceedings of the 2002 IEEE Conference on Decision and Control*, 2002, pp. 22–27.

[17] Bernstein, P. and Goodman, N., "Concurrency Control in Distributed Database Systems," *ACM Computing Surveys*, Vol. 13, No. 2, June 1981, pp. 185–221.

[18] Thomasian, A., "Concurrency Control: Methods, Performance, and Analysis," *ACM Computing Surveys*, Vol. 30, No. 1, March 1998, pp. 70–119.

[19] Devereaux, J., Wohletz, J., Khovanova, T., Vrablik, R., Wu, C., Clarke, J., and Flint, M., "Distributed Cooperative Autonomous Teams (DCAT)," Final report for contract number F33615-02-C-3262, BAE Systems, 2005.